09/807607 PCT/FR99/02172

RÉPUBLIQUE FRANÇAISE

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

REC'D 27 SEP 1999
WIPO PCT

FR 99/2172

EJV

BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 20 SEP. 1999

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

M Houc

Martine PLANCHE

INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE
SIEGE
26 bis, rue de Saint Petersbourg
75800 PARIS Cédex 08
Téléphone : 01 53 04 53 04
Télécopie : 01 42 93 59 30

ETABLISSEMENT PUBLIC NATIONAL CREE PAR LA LOI N° 51-444 DU 19 AVRIL 1951

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

BREVET D'INVENTION, CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle-Livre VI

cerfa N° 55 -1328

REQUÊTE EN DÉLIVRANCE

Confirmation d'un dépôt par télécopie ☐

Cet imprimé est à remplir à l'encre noire en lettres capitales

Réservé à l'INPI

DATE DE REMISE DES PIÈCES 16 OCT. 1998
N° D'ENREGISTREMENT NATIONAL 98 12989
DÉPARTEMENT DE DÉPÔT 75
DATE DE DÉPÔT 16 OCT. 1998

1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE

Cabinet BALLOT-SCHMIT
16, avenue du Pont Royal
94230 Cachan

n°du pouvoir permanent 014238 — références du correspondant — téléphone 01.49.69.91.91

2 DEMANDE Nature du titre de propriété industrielle

☒ brevet d'invention ☐ demande divisionnaire ☐ certificat d'utilité ☐ transformation d'une demande de brevet européen

demande initiale : ☐ brevet d'invention ☐ certificat d'utilité n° date

Établissement du rapport de recherche ☐ différé ☒ immédiat

Le demandeur, personne physique, requiert le paiement échelonné de la redevance ☐ oui ☒ non

Titre de l'invention (200 caractères maximum)

Procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme de cryptographie à clé secrète

3 DEMANDEUR (S) n° SIREN code APE-NAF

Nom et prénoms (souligner le nom patronymique) ou dénomination

GEMPLUS

Forme juridique

S.C.A.
(Société en Commandite par Actions)

Nationalité (s) Française

Adresse (s) complète (s)

Avenue du Pic de Bertagne
Parc d'activités de la Plaine de Jouques
13420 GEMENOS

Pays

FRANCE

En cas d'insuffisance de place, poursuivre sur papier libre ☐

4 INVENTEUR (S) Les inventeurs sont les demandeurs ☐ oui ☐ non Si la réponse est non, fournir une désignation séparée

5 RÉDUCTION DU TAUX DES REDEVANCES ☐ requise pour la 1ère fois ☐ requise antérieurement au dépôt ; joindre copie de la décision d'admission

6 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE

pays d'origine	numéro	date de dépôt	nature de la demande

7 DIVISIONS antérieures à la présente demande n° date n° date

8 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE (nom et qualité du signataire)

BORIN Lydie
Mandataire n° 94-0506
Cabinet BALLOT-SCHMIT

SIGNATURE DU PRÉPOSÉ À LA RÉCEPTION

SIGNATURE APRÈS ENREGISTREMENT DE LA DEMANDE À L'INPI

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

BREVET D'INVENTION, CERTIFICAT D'UTILITE

DÉSIGNATION DE L'INVENTEUR
(si le demandeur n'est pas l'inventeur ou l'unique inventeur)

DEPARTEMENT DES BREVETS

26bis, rue de Saint-Pétersbourg
75800 Paris Cédex 08
Tél. : 01 53 04 53 04 - Télécopie : 01 42 93 59 30

N° D'ENREGISTREMENT NATIONAL

98 12989

n° 014238
TITRE DE L'INVENTION :

Procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme de cryptographie à clé secrète

LE(S) SOUSSIGNÉ(S)

Lydie BORIN

Cabinet BALLOT-SCHMIT
16, avenue du Pont Royal
94230 Cachan

DÉSIGNE(NT) EN TANT QU'INVENTEUR(S) (indiquer nom, prénoms, adresse et souligner le nom patronymique) :

CLAVIER Christophe

BENOIT Olivier

domiciliés au : Cabinet BALLOT-SCHMIT
16, avenue du Pont Royal
94230 Cachan

NOTA : A titre exceptionnel, le nom de l'inventeur peut être suivi de celui de la société à laquelle il appartient (société d'appartenance) lorsque celle-ci est différente de la société déposante ou titulaire.

Date et signature (s) du (des) demandeur (s) ou du mandataire

Cachan, le 16 octobre 1998

L. Borin

BORIN Lydie
Mandataire n° 94-0506
Cabinet BALLOT-SCHMIT

BA 113/140897

PROCÉDÉ DE CONTRE-MESURE DANS UN COMPOSANT ÉLECTRONIQUE METTANT EN OEUVRE UN ALGORITHME DE CRYPTOGRAPHIE A CLÉ SECRETE

La présente invention concerne un procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme de cryptographie à clé secrète. Ils sont utilisés dans des applications où l'accès à des services ou à des données est sévèrement contrôlé. Ils ont une architecture formée autour d'un microprocesseur et de mémoires, dont une mémoire programme qui contient la clé secrète.

Ces composants sont notamment utilisés dans les cartes à puce, pour certaines applications de celles-ci. Ce sont par exemple des applications d'accès à certaines banques de données, des applications bancaires, des applications de télépéage, par exemple pour la télévision, la distribution d'essence ou encore le passage de péages d'autoroutes.

Ces composants ou ces cartes mettent donc en oeuvre un algorithme de cryptographie à clé secrète, dont le plus connu est l'algorithme DES (pour *Data Encryption Standard* dans la littérature anglo-saxonne). D'autres algorithmes à clé secrète existent, comme l'algorithme RC5 ou encore l'algorithme COMP128. Cette liste n'est bien sûr pas exhaustive.

De manière générale et succincte, ces algorithmes ont pour fonction de calculer un message chiffré à partir d'un message appliqué en entrée (à la carte) par un système hôte (serveur, distributeur bancaire...) et de la clé secrète contenue dans la carte, et de fournir en retour au système hôte ce message chiffré, ce qui permet par exemple au système hôte d'authentifier le composant ou la carte, d'échanger des données...

Or il est apparu que ces composants ou ces cartes sont vulnérables à des attaques consistant en une analyse différentielle de consommation en courant et qui permettent à des tiers mal intentionnés de trouver la clé secrète. Ces attaques sont appelées attaques DPA, acronyme anglo-saxon pour *Differential Power Analysis*.

Le principe de ces attaques DPA repose sur le fait que la consommation en courant du microprocesseur exécutant des instructions varie selon la donnée manipulée.

Notamment, une instruction du microprocesseur manipulant un bit de donnée génère deux profils de courant différents selon que ce bit vaut "1" ou "0". Typiquement, si l'instruction manipule un "0", on a à cet instant d'exécution une première amplitude du courant consommé et si l'instruction manipule un "1", on a une deuxième amplitude du courant consommé, différente de la première.

Les caractéristiques des algorithmes de cryptographie sont connues : calculs effectués, paramètres utilisés. La seule inconnue est la clé secrète contenue en mémoire programme. Celle-ci ne peut être déduite de la seule connaissance du message appliqué en entrée et du message chiffré fourni en retour.

Cependant, dans un algorithme de cryptographie, certaines données calculées dépendent seulement du message appliqué en clair en entrée de la carte et de la clé secrète contenue dans la carte. D'autres données calculées dans l'algorithme peuvent aussi être recalculées seulement à partir du message chiffré (généralement fourni en clair en sortie de la carte vers le système hôte) et de la clé secrète contenue dans la carte. Plus précisément, chaque bit de ces données particulières peut être déterminé à partir du

message d'entrée ou de sortie, et d'un nombre limité de bits particuliers de la clé.

Ainsi, à chaque bit d'une donnée particulière, correspond une sous-clé formée par un groupe particulier de bits de la clé.

Les bits de ces données particulières qui peuvent être prédites sont appelés dans la suite, bits cibles.

L'idée de base de l'attaque DPA est ainsi d'utiliser la différence du profil de consommation en courant d'une instruction selon qu'elle manipule un "1" ou un "0" et la possibilité de calculer un bit cible par les instructions de l'algorithme à partir d'un message connu d'entrée ou de sortie et d'une hypothèse sur la sous-clé correspondante.

Le principe de l'attaque DPA est donc de tester une hypothèse de sous-clé donnée, en appliquant sur un grand nombre de courbes de mesure en courant, chacune relative à un message d'entrée connu de l'attaquant, une fonction booléenne de sélection, fonction de l'hypothèse de sous-clé, et définie pour chaque courbe par la valeur prédite pour un bit cible.

En faisant une hypothèse sur la sous-clé concernée, on est en effet capable de prédire la valeur "0" ou "1" que va prendre ce bit cible pour un message d'entrée ou de sortie donné.

On peut alors appliquer comme fonction booléenne de sélection, la valeur prédite "0" ou "1" par le bit cible pour l'hypothèse de sous-clé considérée, pour trier ces courbes en deux paquets : un premier paquet regroupe les courbes qui ont vu la manipulation du bit cible à "0" et un deuxième paquet regroupe les courbes qui ont vu la manipulation du bit cible à "1" selon l'hypothèse de sous-clé. En faisant la moyenne de consommation en courant dans chaque paquet, on obtient une courbe de consommation moyenne M0 (t) pour le

premier paquet et une courbe de consommation moyenne M1(t) pour le deuxième paquet.

Si l'hypothèse de sous-clé est juste, le premier paquet regroupe réellement toutes les courbes parmi les N courbes qui ont vu la manipulation du bit cible à "0" et le deuxième paquet regroupe réellement toutes les courbes parmi les N courbes qui ont vu la manipulation du bit cible à "1". La courbe moyenne de consommation M0(t) du premier paquet aura alors une consommation moyenne partout sauf aux moments de l'exécution des instructions critiques, avec un profil de consommation en courant caractéristique de la manipulation du bit cible à "0" ($profil_0$). En d'autres termes, pour toutes ces courbes tous les bits manipulés ont eu autant de chances de valoir "0" que de valoir "1", sauf le bit cible qui a toujours eu la valeur "0". Ce qui peut s'écrire :

$M0(t)=[(profil_0+profil_1)/2]_{t\neq tci} + [profil_0]_{tci}$ soit

$M0(t)=[Vm_t]_{t\neq tci} + [profil_0]_{tci}$

où tci représente les instants critiques, auxquels une instruction critique a été exécutée.

De même, la courbe moyenne de consommation M1(t) du deuxième paquet correspond à une consommation moyenne partout sauf aux moments de l'exécution des instructions critiques, avec un profil de consommation en courant caractéristique de la manipulation du bit cible à "1" ($profil_1$). On peut écrire :

$M1(t)=[(profil_0+profil_1)/2]_{t\neq tci} + [profil_1]_{tci}$ soit

$M1(t)=[Vm_t]_{t\neq tci} + [profil_1]_{tci}$

On a vu que les deux profils $profil_0$ et $profil_1$ ne sont pas égaux. La différence des courbes M0(t) et M1(t) donne alors un signal DPA(t) dont l'amplitude est égale à $profil_0$-$profil_1$ aux instants critiques tci d'exécution des instructions critiques manipulant ce bit, c'est à dire, dans l'exemple représenté sur la figure 1, aux endroits tc0 à tc6, et dont l'amplitude

est à peu près égale à zéro en dehors des instants critiques.

Si l'hypothèse de sous-clé est fausse, le tri ne correspond pas à la réalité. Statistiquement, il y a alors dans chaque paquet, autant de courbes ayant vu réellement la manipulation du bit cible à "0" que de courbes ayant vu la manipulation du bit cible à "1". La courbe moyenne résultante M0(t) se situe alors autour d'une valeur moyenne donnée par $(profil_0+profil_1)/2=Vm$, car pour chacune des courbes, tous les bits manipulés, y compris le bit cible ont autant de chances de valoir "0" que de valoir "1".

Le même raisonnement sur le deuxième paquet conduit à une courbe moyenne de consommation en courant M1(t) dont l'amplitude se situe autour d'une valeur moyenne donnée par $(profil_0+profil_1)/2=Vm$.

Le signal DPA(t) fourni par la différence M0(t)-M1(t) est dans ce cas sensiblement égal à zéro. Le signal DPA(t) dans le cas d'une hypothèse de sous-clé fausse est représenté sur la figure 2.

Ainsi l'attaque DPA exploite la différence du profil de consommation en courant pendant l'exécution d'une instruction suivant la valeur du bit manipulé, pour effectuer un tri de courbes de consommation en courant selon une fonction de sélection booléenne pour une hypothèse de sous-clé donnée. En effectuant une analyse différentielle de la consommation moyenne en courant entre les deux paquets de courbes obtenus, on obtient un signal d'information DPA(t).

Le déroulement d'une attaque DPA consiste alors globalement:

a- à tirer N messages aléatoires (par exemple N égal 1000);

b- à faire exécuter l'algorithme par la carte pour chacun des N messages aléatoires, en relevant la courbe

de consommation en courant à chaque fois (mesurée sur la borne d'alimentation du composant);

c- à faire une hypothèse sur une sous-clé;

d- à prédire, pour chacun des messages aléatoires, la valeur prise par un des bits cibles dont la valeur ne dépend que des bits du message (d'entrée ou de sortie) et de la sous-clé prise en hypothèse, pour obtenir la fonction de sélection booléenne;

e- à trier les courbes selon cette fonction de sélection booléenne (c'est à dire selon la valeur "0" ou "1" prédite pour ce bit cible pour chaque courbe sous l'hypothèse de sous-clé);

f- à calculer dans chaque paquet la courbe résultante de consommation moyenne en courant;

g- à effectuer la différence de ces courbes moyennes, pour obtenir le signal DPA(t).

Si l'hypothèse sur la sous-clé est juste, la fonction de sélection booléenne est juste et les courbes du premier paquet correspondent réellement aux courbes pour lesquelles le message appliqué en entrée ou en sortie a donné un bit cible à "0" dans la carte et les courbes du deuxième paquet correspondent réellement aux courbes pour lesquelles le message appliqué en entrée ou en sortie a donné un bit cible à "1" dans la carte.

On est dans le cas de la figure 1 : le signal DPA(t) n'est donc pas nul aux instants tc0 à tc6 correspondant à l'exécution des instructions critiques (celles qui manipulent le bit cible).

On notera que l'attaquant n'a pas besoin de connaître avec précision les instants critiques. Il suffit qu'il y ait au moins un instant critique dans la période d'acquisition.

Si l'hypothèse de sous-clé n'est pas juste, le tri ne correspond pas à la réalité et on a alors dans chaque paquet autant de courbes correspondant en

réalité à un bit cible à "0" que de courbes correspondant à un bit cible à "1". Le signal DPA(t) est sensiblement nul partout (cas représenté à la figure 2). Il faut retourner à l'étape c- et faire une nouvelle hypothèse sur la sous-clé.

Si l'hypothèse s'avère juste, on peut passer à l'évaluation d'autres sous-clés, jusqu'à avoir reconstitué la clé au maximum. Par exemple, avec un algorithme DES, on utilise une clé de 64 bits, dont seulement 56 bits utiles. Avec une attaque DPA, on est capable de reconstituer au moins 48 bits des 56 bits utiles.

La présente invention a pour but de mettre en oeuvre dans un composant électronique, un procédé de contre-mesure qui entraîne un signal DPA(t) nul, même dans le cas où l'hypothèse de sous-clé est juste.

De cette façon, rien ne permet de distinguer le cas de l'hypothèse de sous-clé juste des cas d'hypothèses de sous-clé fausses. Par cette contre-mesure, le composant électronique est paré contre les attaques DPA.

Selon l'invention, le procédé de contre-mesure permet de rendre imprédictibles les bits cibles, c'est à dire les données manipulées par des instructions critiques.

En effet, du fait de la contre-mesure, pour chaque message appliqué en entrée, un bit cible manipulé par une instruction critique prend la valeur 0 ou 1 avec une égale probabilité. Dans chaque paquet de courbes que fera l'attaquant sous une hypothèse de sous-clé donnée, au moyen de la fonction de sélection booléenne qu'il aura calculée, on aura autant de courbes ayant réellement manipulé un bit cible "0" que de courbes ayant réellement manipulé un bit cible à "1". Le signal DPA(t) sera toujours nul, que l'hypothèse de sous-clé soit juste ou non.

Telle que caractérisée, l'invention concerne donc un procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme cryptographique à clé secrète, la mise en oeuvre de l'algorithme comprenant l'utilisation de premiers moyens pour fournir une donnée de sortie à partir d'une donnée d'entrée, la donnée de sortie et/ou des données dérivées étant manipulées par des instructions critiques. Selon l'invention, le procédé de contre-mesure prévoit l'utilisation d'autres moyens, en sorte que la donnée de sortie et les données dérivées soient imprédictibles.

Selon l'invention, l'utilisation des différents moyens est gérée selon une loi statistique de probabilité un demi.

D'autres caractéristiques et avantages de l'invention sont détaillés dans la description suivante faite à titre indicatif et nullement limitatif et en référence aux dessins annexés, dans lesquels :

- les figures 1 et 2 déjà décrites représentent le signal DPA(t) que l'on peut obtenir en fonction d'une hypothèse sur une sous-clé de la clé secrète K, selon une attaque DPA;
- les figures 3 et 4 sont des organigrammes d'exécution des premiers tours et derniers tours de l'algorithme DES;
- la figure 5 est un schéma-bloc de l'opération SBOX utilisée dans l'algorithme DES;
- la figure 6 montre un exemple de table de constante élémentaire à une entrée et une sortie utilisée dans l'opération SBOX;
- les figures 7 et 8 montrent un exemple d'organigramme d'exécution des premiers et derniers tours de l'algorithme DES, selon un mode de réalisation du procédé de contre-mesure selon l'invention;

- les figures 9 et 10 montrent respectivement une deuxième et une troisième tables de constantes élémentaires selon l'invention;
- la figure 11 représente un organigramme général d'exécution du DES selon un mode de réalisation du procédé de contre-mesure selon l'invention; et
- la figure 12 représente un schéma-bloc simplifié d'une carte à puce comportant un composant électronique dans lequel le procédé de contre-mesure selon l'invention est mis en oeuvre.

La présente invention va être expliquée dans un exemple d'application à l'algorithme cryptographique DES. L'invention n'est pas limitée à ce seul exemple. Elle s'applique aux algorithmes cryptographiques à clé secrète en général.

L'algorithme cryptographique DES (dans la suite on parlera plus simplement du DES ou de l'algorithme DES) comporte 16 tours de calcul, notés T1 à T16, comme représenté sur les figures 3 et 4.

Le DES débute par une permutation initiale IP sur le message d'entrée M (figure 3). Le message d'entrée M est un mot f de 64 bits. Après permutation, on obtient un mot e de 64 bits, que l'on coupe en deux pour former les paramètres d'entrée L0 et R0 du premier tour (T1). L0 est un mot d de 32 bits contenant les 32 bits de poids forts du mot e. R0 est un mot h de 32 bits contenant les 32 bits de poids faibles du mot e.

La clé secrète K, qui est un mot q de 64 bits subit elle-même une permutation et une compression pour fournir un mot r de 56 bits.

Le premier tour comprend une opération EXP PERM sur le paramètre R0, consistant en une expansion et une permutation, pour fournir en sortie un mot l de 48 bits.

Ce mot l est combiné à un paramètre K1, dans une opération de type OU EXCLUSIF notée XOR, pour fournir un mot b de 48 bits. Le paramètre K1 qui est un mot m de 48 bits est obtenu du mot r par un décalage d'une position (opération notée SHIFT sur les figures 3 et 4) suivi d'une permutation et d'une compression (opération notée COMP PERM).

Le mot b est appliqué à une opération notée SBOX, en sortie de laquelle on obtient un mot a de 32 bits. Cette opération particulière sera expliquée plus en détail en relation avec les figures 5 et 6.

Le mot a subit une permutation P PERM, donnant en sortie le mot c de 32 bits.

Ce mot c est combiné au paramètre d'entrée L0 du premier tour T1, dans une opération logique de type OU EXCLUSIF, notée XOR, qui fournit en sortie le mot g de 32 bits.

Le mot h (=R0) du premier tour fournit le paramètre d'entrée L1 du tour suivant (T2) et le mot g du premier tour fournit le paramètre d'entrée R1 du tour suivant. Le mot p du premier tour fournit l'entrée r du tour suivant.

Les autres tours T2 à T16 se déroulent de façon similaire, excepté en ce qui concerne l'opération de décalage SHIFT qui se fait sur une ou deux positions selon les tours considérés.

Chaque tour Ti reçoit ainsi en entrée les paramètres Li-1, Ri-1 et r et fournit en sortie les paramètres Li, Ri et r pour le tour suivant Ti+1.

En fin d'algorithme DES (figure 4), le message chiffré est calculé à partir des paramètres L16 et R16 fournis par le dernier tour T16.

Ce calcul du message chiffré C comprend en pratique les opérations suivantes :

- formation d'un mot e' de 64 bits en inversant la position des mots L16 et R16, puis en les concaténant;

- application de la permutation IP^{-1} inverse de celle de début de DES, pour obtenir le mot f' de 64 bits formant le message chiffré C.

L'opération SBOX est détaillée sur les figures 5 et 6. Elle comprend une table de constantes TC_0 pour fournir une donnée de sortie a en fonction d'une donnée d'entrée b.

En pratique, cette table de constantes TC_0 se présente sous la forme de huit tables de constantes élémentaires TC_01 à TC_08, chacune recevant en entrée seulement 6 bits du mot b, pour fournir en sortie seulement 4 bits du mot a.

Ainsi, la table de constante élémentaire TC_01 représentée sur la figure 6 reçoit comme donnée d'entrée, les bits b1 à b6 du mot b et fournit comme donnée de sortie les bits a1 à a4 du mot a.

En pratique ces huit tables de constantes élémentaires sont mémorisées en mémoire programme du composant électronique.

Dans l'opération SBOX du premier tour T1, un bit particulier de la donnée a de sortie de la table de constante TC_0 dépend de seulement 6 bits de la donnée b appliquée en entrée, c'est à dire de seulement 6 bits de la clé secrète K et du message d'entrée (M).

Dans l'opération SBOX du dernier tour T16, un bit particulier de la donnée a de sortie de la table de constante TC_0 peut être recalculé à partir de seulement 6 bits de la clé secrète K et du message chiffré (C).

Or si on reprend le principe de l'attaque DPA, si on choisit comme bit cible un bit de la donnée de sortie a, il suffit de faire une hypothèse sur 6 bits de la clé K, pour prédire la valeur d'un bit cible pour un message d'entrée (M) ou de sortie (C) donné. En d'autres termes, pour le DES, il suffit de faire une hypothèse sur une sous-clé de 6 bits.

Dans une attaque DPA sur un tel algorithme pour un bit cible donné, on a donc à discriminer une hypothèse de sous-clé juste parmi 64 possibles.

Ainsi, en prenant seulement huit bits du mot a comme bits cibles, (un bit de sortie par table de constantes élémentaire $TC_0 1$ à $TC_0 8$), on peut découvrir jusqu'à 6x8=48 bits de la clé secrète, en faisant des attaques DPA sur chacun de ces bits cibles.

Dans le DES, on trouve donc des instructions critiques au sens des attaques DPA au début de l'algorithme et à la fin.

Au début de l'algorithme DES, les données qui peuvent être prédites à partir d'un message d'entrée M et d'une hypothèse de sous-clé, sont les données a et g calculées dans le premier tour (T1).

La donnée a du premier tour T1 (figure 3) est la donnée de sortie de l'opération SBOX du tour considéré. La donnée g est calculée à partir de la donnée a, par permutation (P PERM) et opération OU EXCLUSIF avec le paramètre d'entrée L0.

En fait, la donnée c du premier tour, est une donnée dérivée de la donnée a du premier tour. La donnée dérivée c correspond à une simple permutation de bits de la donnée a.

La donnée l du deuxième tour est une donnée dérivée de la donnée g du premier tour, car elle correspond à une permutation des bits du mot g, certains bits du mot g étant en outre dupliqués.

Connaissant a et g, on peut aussi connaître ces données dérivées.

Les instructions critiques du début de l'algorithme sont les instructions critiques qui manipulent soit la donnée que l'on peut prédire, comme la donnée a du premier tour, soit une donnée dérivée.

Les instructions critiques manipulant la donnée a du premier tour T1 ou la donnée dérivée c sont ainsi

les instructions de fin de l'opération SBOX, de l'opération P PERM et de début de l'opération XOR du premier tour T1.

Les instructions critiques manipulant la donnée g ou des données dérivées sont toutes les instructions de fin d'opération XOR de fin du premier tour T1 jusqu'aux instructions de début d'opération SBOX du deuxième tour T2, et les instructions de début d'opération XOR de fin du troisième tour T3 (L2 = h(T2) = g(T1)).

En fin d'algorithme DES, les données qui peuvent être prédites à partir d'un message chiffré C et d'une hypothèse de sous-clé, sont la donnée a du seizième tour T16 et la donnée L15 égale au mot h du quatorzième tour T14.

Les instructions critiques manipulant la donnée a du seizième tour ou des données dérivées sont les instructions du seizième tour de fin d'opération SBOX, de l'opération de permutation P PERM et de début d'opération XOR.

Pour la donnée L15, les instructions critiques manipulant cette donnée ou des données dérivées sont toutes les instructions depuis les instructions de fin d'opération XOR du quatorzième tour T14, jusqu'aux instructions de début d'opération SBOX du quinzième tour T15, et les instructions de début d'opération XOR de fin du seizième tour T16.

Le procédé de contre-mesure selon l'invention appliqué à cet algorithme DES consiste à avoir, pour chaque instruction critique, autant de chances que l'instruction critique manipule une donnée que son complément. Ainsi, quel que soit le bit cible sur lequel l'attaque DPA peut être faite, on a autant de chances que les instructions critiques qui manipulent ce bit, manipulent un "1" ou un "0".

En pratique, ceci doit être vrai pour chacun des bits cibles potentiels : en d'autres termes,

l'attaquant ayant le choix entre plusieurs attaques possibles, c'est à dire entre plusieurs fonctions de sélection booléenne possibles pour effectuer son tri de courbes, pour une hypothèse de sous-clé donnée, la mise en oeuvre du procédé de contre-mesure selon l'invention doit s'attacher à ce que les données manipulées par chacune des instructions critiques, prennent aléatoirement, une fois sur deux, une valeur ou son complément. En ce qui concerne l'application du procédé de contre-mesure selon l'invention à l'algorithme DES, il faut donc appliquer la contre-mesure aux instructions critiques de début de DES et aux instructions critiques de fin de DES, pour être totalement protégé.

Dans le DES, toutes les données manipulées par des instructions critiques sont une donnée de sortie ou des données dérivées d'une donnée de sortie d'une opération SBOX.

En effet, en début de DES, les données qui peuvent être prédites sont les données a et g du premier tour T1. La donnée a est la donnée de sortie de l'opération SBOX du premier tour. La donnée g est calculée à partir de la donnée a, puisque g = P PERM(a) XOR L0. g est donc une donnée dérivée de la donnée de sortie a de l'opération SBOX du premier tour. Ainsi, toutes les données manipulées par les instructions critiques de début de DES découlent directement ou indirectement de la donnée de sortie a de l'opération SBOX du premier tour.

En ce qui concerne la fin de DES, les données qui peuvent être prédites sont la donnée a du seizième tour T16 et la donnée g du quatorzième tour T14, g étant égale à L15.

La donnée a est la donnée de sortie de l'opération SBOX du seizième tour T16.

Quant à la donnée L15, elle se calcule, dans l'exécution normale de l'algorithme DES, à partir de la donnée de sortie a de l'opération SBOX du quatorzième tour T14 : L15 = P PERM(a) XOR L14.

Si on rend imprédictibles les données de sortie a de ces opérations SBOX particulières, on rend aussi imprédictibles toutes les données dérivées : on rend donc imprédictibles toutes les données manipulées par les instructions critiques de l'algorithme DES.

L'opération SBOX correspond donc à des premiers moyens , qui consistent en une table de constantes TC_0, et qui sont utilisés dans chaque tour pour fournir une donnée de sortie E à partir d'une donnée d'entrée S.

Un mode de réalisation du procédé de contre-mesure appliqué à l'algorithme DES peut consister à utiliser au moins une autre table de constantes comme autres moyens pour rendre imprédictible la donnée de sortie a, en sorte que cette donnée de sortie et/ou des données dérivées manipulées par les instructions critiques soient toutes imprédictibles.

Dans l'exécution de l'algorithme, l'utilisation des diiférents moyens, c'est à dire, dans l'exemple, des différentes tables de constantes est gérée selon une loi statistique de probabilité un demi.

L'autre table de constantes ou les autres tables de constantes sont telles qu'à l'une et/ou l'autre des données d'entrée d et de sortie de la première table de constantes TC_0, elles font correspondre la donnée complémentée.

Les figures 7 et 8 représentent ainsi un mode d'application du procédé de contre-mesure de l'invention appliqué à l'algorithme DES.

La figure 7 représente le début de l'algorithme. Les opérations et données non modifiées par le procédé de contre-mesure selon l'invention portent les mêmes références que dans la figure 3 déjà décrite.

En début d'algorithme DES, on prévoit une deuxième table de constante TC_1 dans l'opération SBOX du premier tour T1. Toutes les données affectées par cette deuxième table de constantes TC_1 sont affectées d'un signe ' ou d'un signe $^-$ sur ces figures. On voit que les instructions critiques de début de DES manipulent toutes des données affectées par le procédé de contre-mesure.

On remarquera que la première table de constantes étant en fait formée de huit premières tables de constantes, la deuxième table de constantes est également formée de huit deuxièmes tables de constantes.

Dans l'exemple de réalisation représenté, la première table de constantes TC_0 et la deuxième table de constantes TC_1 sont telles que pour une même donnée d'entrée E, la deuxième fournit en sortie le complément /S de la donnée de sortie S fournie par la première.

La figure 9 montre une telle deuxième table élémentaire $TC_1 1$ fournissant une sortie complémentée par rapport à la première table élémentaire $TC_0 1$ montrée sur la figure 6.

Avec une telle deuxième table de constantes TC_1, on obtient, en sortie de l'opération SBOX du premier tour T1, le complément /a de la donnée a obtenue avec la première table de constante TC_0. De même, on obtient dans le premier tour T1 la donnée complémentée /g et dans le deuxième tour T2, les données complémentées /h, /L2, /l et /b.

En utilisant la première table ou la deuxième table pour fournir la donnée de sortie selon une loi statistique de probabilité un demi, tous les bits de cible potentiels de début de DES manipulés par les instructions critiques ont autant de chances de prendre la valeur "1" que de prendre la valeur "0".

En fin d'algorithme DES, le mode de réalisation du procédé de contre-meseure selon l'invention nécessite l'utilisation de plusieurs tables de constantes différentes de la première, car il faut considérer à la fois la donnée de sortie a calculée au quatorzième tour T14, et la donnée de sortie a calculée au seizième tour T16 pour rendre imprédictibles toutes les données manipulées par les instructions critiques de cette fin de DES.

Un exemple de réalisation du procédé de contre-mesure appliqué à cette fin d'algorithme DES est représenté sur la figure 8.

Il prévoit l'utilisation de deux tables de constantes TC_1 et TC_2.

Dans l'opération SBOX du quatorzième tour T14, on utilise la deuxième table de constantes TC_1 déjà utilisée pour le début du DES.

Et on utilise une troisième table de constantes TC_2, dans les opérations SBOX des quinzième et seizième tours.

Cette troisième table de constantes TC_2 est telle qu'elle fournit le complément /S de la donnée de sortie S au complément /E de la donnée d'entrée E de la première table de constantes TC_0. Un exemple d'une troisième table de constantes élémentaire TC_21 correspondante, à partir de la première table de constantes élémentaire TC_0 est montré sur la figure 10.

En utilisant de telles tables de constantes, il apparaît sur la figure 8 que toutes les instructions critiques manipulent des données complémentées.

L'invention ne se limite pas à ces seuls exemples de tables de constantes TC_1 et TC_2. D'autres possibilités existent. Par exemple, pour le procédé de contre-mesure appliqué à la fin de DES, il est aussi possible de combiner l'utilisation de la table de constantes TC_1 avec une autre table de constantes

définie par rapport à la première table de constantes TC_0 comme fournissant la donnée de sortie S au complément /E de la donnée d'entrée.

D'une manière générale, la fin de DES nécessite l'utilisation de différentes tables de constantes, en fonction des tours considérés, pour que toutes les données manipulées par les instructions critiques de cette fin de DES soient imprédictibles.

Le mode de réalisation décrit en relation avec les figures 7 et 8 a cependant un inconvénient : le procédé de contre-mesure appliqué en entrée de DES produit des résultats intermédiaires calculés L3' et R3' qui ne sont pas justes. Tous les résultats intermédiaires suivants ne sont donc pas justes non plus.

De même, en fin de DES, le procédé de contre-mesure appliqué en entrée de DES produit des résultats intermédiaires calculés L16' et R16' qui ne sont pas justes.

Dans tous les cas, le message chiffré est faux.

Dans ce mode de réalisation de l'invention, il faut donc prévoir de pouvoir reprendre à chaque fois la suite de l'algorithme avec les bons résultats intermédiaires, une fois les instructions critiques passées.

En pratique, comme on a vu que les instructions critiques de début de DES se trouvent dans les trois premiers tours, on va dédoubler ces trois premiers tours. En d'autres termes, on prévoit d'exécuter deux séquences comprenant chacune les trois premiers tours T1, T2, T3 au moins. Une première séquence SEQA utilise la première table de constantes TC_0 dans chaque tour. L'autre séquence SEQB utilise la deuxième table de constantes TC_1 au moins dans le premier tour T1. Dans l'exemple représenté, on utilise la première table de constantes dans les deux tours suivants T2 et T3.

On a vu que dans le procédé de contre-mesure selon l'invention, l'utilisation des différents moyens, c'est à dire, dans l'exemple, l'utilisation des différentes tables de constantes, est gérée selon une loi statistique de probabilité un demi. Cette loi statistique de probabilité un demi est alors plus particulièrement appliquée à l'ordre d'utilisation de ces différents moyens, c'est à dire, dans l'exemple, à l'ordre d'éxécution des deux séquences SEQA et SEQB.

De même, pour avoir les bons paramètres L16 et R16 en fin de DES pour élaborer le message chiffré $\bar{C}$, on dédouble également les trois tours T14, T15 et T16 (figure 7) qui contiennent les instructions critiques de fin de DES. On va donc exécuter deux séquences qui comprennent au moins les trois derniers tours T14, T15, T16. Une première séquence SEQA' utilise dans chaque tour la première table de constantes TC_0. L'autre séquence SEQB' utilise les autres tables de constantes TC_1 et TC_2. Comme précédemment, la loi statistique de probabilité un demi est alors appliquée à l'ordre d'exécution de ces deux séquences SEQA' et SEQB'.

Les instructions critiques sont alors exécutées deux fois, une dans chaque séquence. Mais au moment de l'exécution de n'importe laquelle des instructions critiques de l'une ou l'autre des séquences, la probabilité de manipuler une donnée sera égale à la probabilité de manipuler son complément.

Le programme de calcul du DES mis en oeuvre dans le composant électronique doit donc être modifié pour inclure le procédé de contre-mesure selon l'invention. Un exemple d'organigramme d'exécution conforme à l'invention et mettant en oeuvre le procédé de contre-mesure en début et en fin de DES selon le mode de réalisation décrit en relation avec les figures 7 et 8 est représenté sur la figure 11. Dans cet exemple, les séquences SEQA et SEQB comprennent les trois premiers

tours et les séquences SEQA' et SEQB' comprennent les trois derniers tours.

Le programme de calcul consiste alors principalement, au début du calcul, à sauvegarder les paramètres d'entrée notés DATAIN et KEY, qui correspondent en pratique aux paramètres L0, R0 et r, dans une zone mémoire temporaire notée CONTEXT0.

Selon ce programme de calcul, on positionne ensuite un premier compteur de boucle FR à 0, et on tire aléatoirement une valeur RND1 égale à 0 ou à 1.

Si RND1 vaut 1, dans l'exemple, on effectue d'abord la séquence SEQB de T1, T2, T3, dans laquelle on utilise la deuxième table de constantes TC_1 au tour T1 et la première table TC_0 pour les tours T2 et T3. On sauvegarde les paramètres de sortie L3', R3' (qui ont des valeurs fausses) dans une zone mémoire temporaire notée CONTEXT2.

Si FR n'est pas égal à 1, on le met à 1, on restaure les paramètres d'entrée du CONTEXT0 et on complémente la valeur de RND1. Dans l'exemple, on obtient RND1=0. On va alors exécuter l'autre séquence SEQA de T1, T2, T3 dans laquelle on utilise la première table de constante dans les trois tours T1, T2 et T3. On sauvegarde les paramètres de sortie (valeurs justes) dans une zone mémoire temporaire notée CONTEXT1.

Si FR est à 1, c'est que l'on a effectué les deux séquences. On restaure alors CONTEXT1 pour fournir les résultats intermédiaires L3, R3 ayant les valeurs justes, au tour suivant (T4).

Si RND1 vaut zéro, on commence par $T1(TC_0)$, $T2(TC_0)$, $T3(TC_0)$ et on finit par $T1(TC_1)$, $T2(TC_0)$, $T3(TC_0)$.

Arrivé à la fin du tour T13, on sauvegarde les paramètres fournis par ce tour, L13, R13, dans la mémoire temporaire CONTEXT0, et on procède pour les

tours restants T14, T15 et T16 de façon similaire aux premiers tours.

Dans tous les cas, il faut que le nombre d'instructions soit exactement le même quel que soit le chemin de calcul. C'est pour cela notamment que dans l'exemple d'application décrit, on prévoit de sauvegarder aussi les valeurs fausses (L3', R3' ou L16', R16') dans la zone mémoire temporaire CONTEXT2.

En effet si une différence quelconque existait entre les deux chemins possibles, il y aurait alors une possibilité d'attaque DPA fructueuse.

Le procédé de contre-mesure selon l'invention n'est pas limité à l'exemple particulier de réalisation décrit en référence à l'algorithme DES. Il s'applique à tout algorithme de cryptographie à clé secrète. De manière générale, pour toute mise en oeuvre d'un algorithme comprenant l'utilisation de premiers moyens pour fournir une donnée de sortie à partir d'une donnée d'entrée, la donnée de sortie et/ou des données dérivées étant manipulées par des instructions critiques, le procédé de contre-mesure selon l'invention comprend l'utilisation d'autres moyens, en sorte que la donnée de sortie et les données dérivées soient imprédictibles.

L'utilisation des différents moyens , c'est à dire des premiers moyens et des autres moyens, est gérée selon une loi statistique de probabilité un demi.

Les autres moyens peuvent comprendre plusieurs moyens différents. Ils sont tels qu'à l'une ou à l'autre des données d'entrée et de sortie des premiers moyens, ils font correspondre la donnée complémentée.

Dans l'exemple d'un mode d'application du procédé de contre-mesure au DES plus particulièrement décrit, les premiers moyens consistent en la première table de constantes TC_0. Les autres moyens consistent, en début de DES, dans la deuxième table de constantes TC_1. En

fin de DES, ils consistent en deux tables de constantes différentes, TC_1 et TC_2 dans l'exemple.

Pour appliquer le procédé de contre-mesure selon l'invention à un algorithme de cryptographie à clé secrète donné, il faut donc d'abord déterminer toutes les données de cet algorithme qui peuvent être prédites et toutes les instructions critiques au sens de l'attaque DPA manipulant ces données ou des données dérivées. Il faut ensuite identifier dans l'algorithme des premiers moyens et des autres moyens au sens de l'invention, en sorte que toutes les données manipulées par les instructions critiques soient imprédictibles. Les premiers moyens sont, pour l'algorithme DES, la table de constantes TC_0. Les autres moyens sont dans l'exemple, d'autres tables de constantes. Ces moyens peuvent être des opérations différentes pour d'autres algorithmes. Pour un même algorithme, ces moyens peuvent consister en des opérations différentes selon les instructions critiques identifiées.

Le composant électronique 1 mettant en oeuvre un tel procédé de contre-mesure dans un algorithme de cryptographie à clé secrète comprend typiquement, comme représenté sur la figure 12, un microprocesseur µP, une mémoire programme 2 et une mémoire de travail 3. Pour pouvoir gérer l'utilisation des différents moyens selon l'invention, qui sont, dans l'exemple décrit, les différentes tables de constantes mémorisées en mémoire programme, des moyens 4 de génération d'une valeur aléatoire entre 0 et 1, sont prévus qui, si on se reporte à la figure 11, fourniront la valeur de RND1 à chaque exécution du DES. Un tel composant peut tout particulièrement être utilisé dans une carte à puce CP, pour améliorer leur inviolabilité.

REVENDICATIONS

1. Procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme cryptographique à clé secrète (K), la mise en oeuvre de l'algorithme comprenant l'utilisation de premiers moyens (TC_0) pour fournir une donnée de sortie (S) à partir d'une donnée d'entrée (E), la donnée de sortie et/ou des données dérivées étant manipulées par des instructions critiques, caractérisé en ce que le procédé de contre-mesure prévoit l'utilisation d'autres moyens (TC_1), en sorte que la donnée de sortie et les données dérivées soient imprédictibles.

2. Procédé de contre-mesure selon la revendication 1, caractérisé en ce que l'utilisation des différents moyens (TC_0, TC_1) est gérée par une loi statistique de probabilité un demi.

3. Procédé de contre-mesure selon la revendication 2, la mise en oeuvre de l'algorithme comprenant seize tours de calcul (T1, ...,T16), caractérisé en ce qu'il comprend l'exécution d'une première séquence (SEQA) et d'une deuxième séquence (SEQB) formées des trois premiers tours au moins (T1, T2, T3), l'ordre d'exécution des séquences étant fonction de la loi statistique de probabilité un demi, la première séquence (SEQA) utilisant les premiers moyens (TC_0) dans chaque tour, la deuxième séquence (SEQB) utilisant les autres moyens (TC_1) dans le premier tour (T1) au moins.

4. Procédé de contre-mesure selon la revendication 3, caractérisé en ce que la première et la deuxième

séquences sont formées chacune des trois premiers tours (T1, T2, T3).

5. Procédé de contre-mesure selon la revendication 3 ou 4, caractérisé en ce que les autres moyens consistent en des deuxièmes moyens (TC_1) tels que pour une même donnée d'entrée (E), ils fournissent en sortie le complément (/S) de la donnée de sortie (S) des premiers moyens (TC_0).

6. Procédé de contre-mesure selon la revendication 2, la mise en oeuvre de l'algorithme comprenant seize tours de calcul (T1, ...,T16), caractérisé en ce qu'il comprend l'exécution d'une première séquence (SEQA') et d'une deuxième séquence (SEQB') formées chacune des trois derniers tours (T14, T15, T16) au moins, l'ordre d'exécution des séquences étant fonction de la loi statistique de probabilité un demi, la première séquence (SEQA') utilisant les premiers moyens (TC_0) dans chaque tour, la deuxième séquence (SEQB') utilisant les autres moyens (TC_1, TC_2).

7. Procédé de contre-mesure selon la revendication 6, caractérisé en ce que la première et la deuxième séquences sont formées chacune des trois derniers tours, et en ce que les autres moyens utilisés dans la deuxième séquence comprennent des deuxièmes moyens (TC_1) et des troisièmes moyens (TC_2).

8. Procédé de contre-mesure selon la revendication 6 ou 7, caractérisé en ce que les deuxièmes moyens (TC_1) sont tels que pour une même donnée d'entrée (E), ils fournissent en sortie le complément (/S) de la donnée de sortie (S) des premiers moyens (TC_0) et en ce que ces deuxièmes moyens sont utilisés dans la deuxième séquence (SEQB') pour le quatorzième tour (T14).

9. Procédé de contre-mesure selon la revendication 8, caractérisé en ce que les troisièmes moyens (TC_2) sont tels que pour le complément de la donnée d'entrée (E), ils fournissent en sortie le complément (/S) de la donnée de sortie (S) des premiers moyens (TC_0) et sont utilisés dans la deuxième séquence, pour le quinzième tour et le seizième tour (T15, T16).

10. Procédé de contre-mesure selon l'une quelconque des revendications précédentes, caractérisé en ce que les différents moyens sont des tables de constantes.

11. Composant électronique mettant en oeuvre le procédé de contre-mesure selon l'une quelconque des revendications précédentes, caractérisé en ce que les différents moyens (TC_0, TC_1, TC_2) pour fournir une donnée de sortie à partir d'une donnée d'entrée sont fixés en mémoire programme du dit composant et en ce qu'il comprend des moyens de génération d'une valeur aléatoire (RND1) à 0 ou à 1 pour gérer l'utilisation des dits différents moyens.

12. Carte à puce comprenant un composant électronique selon la revendication 11.

FIG.1

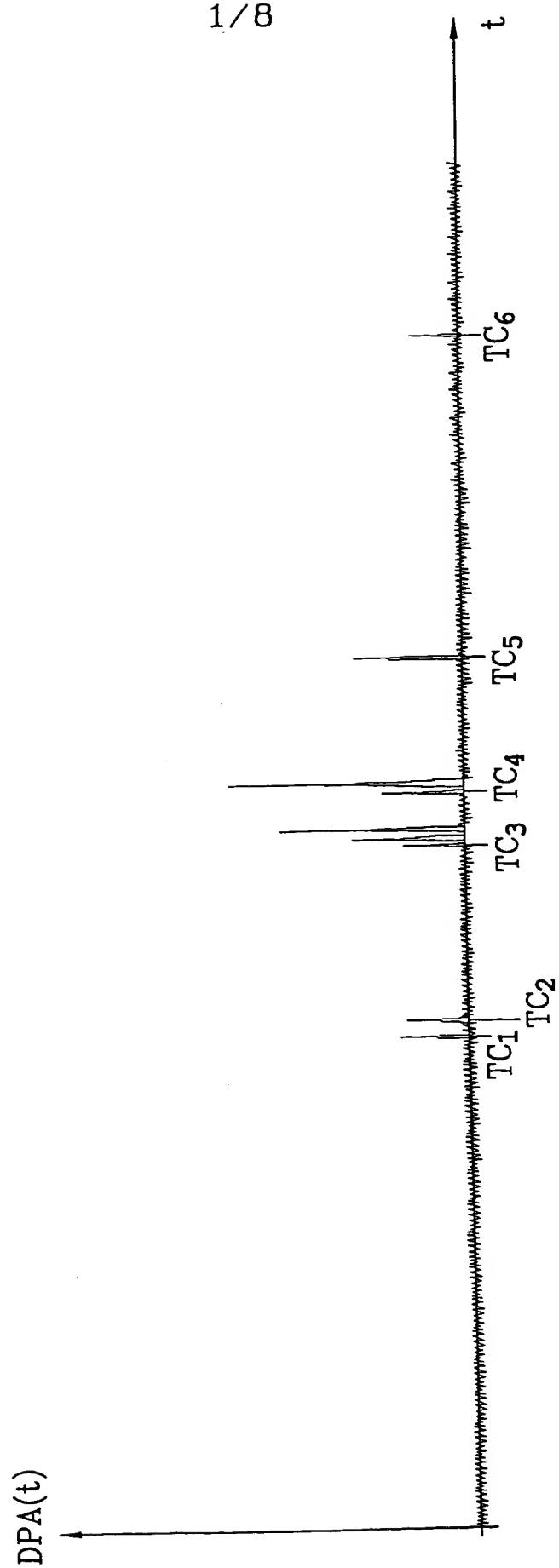
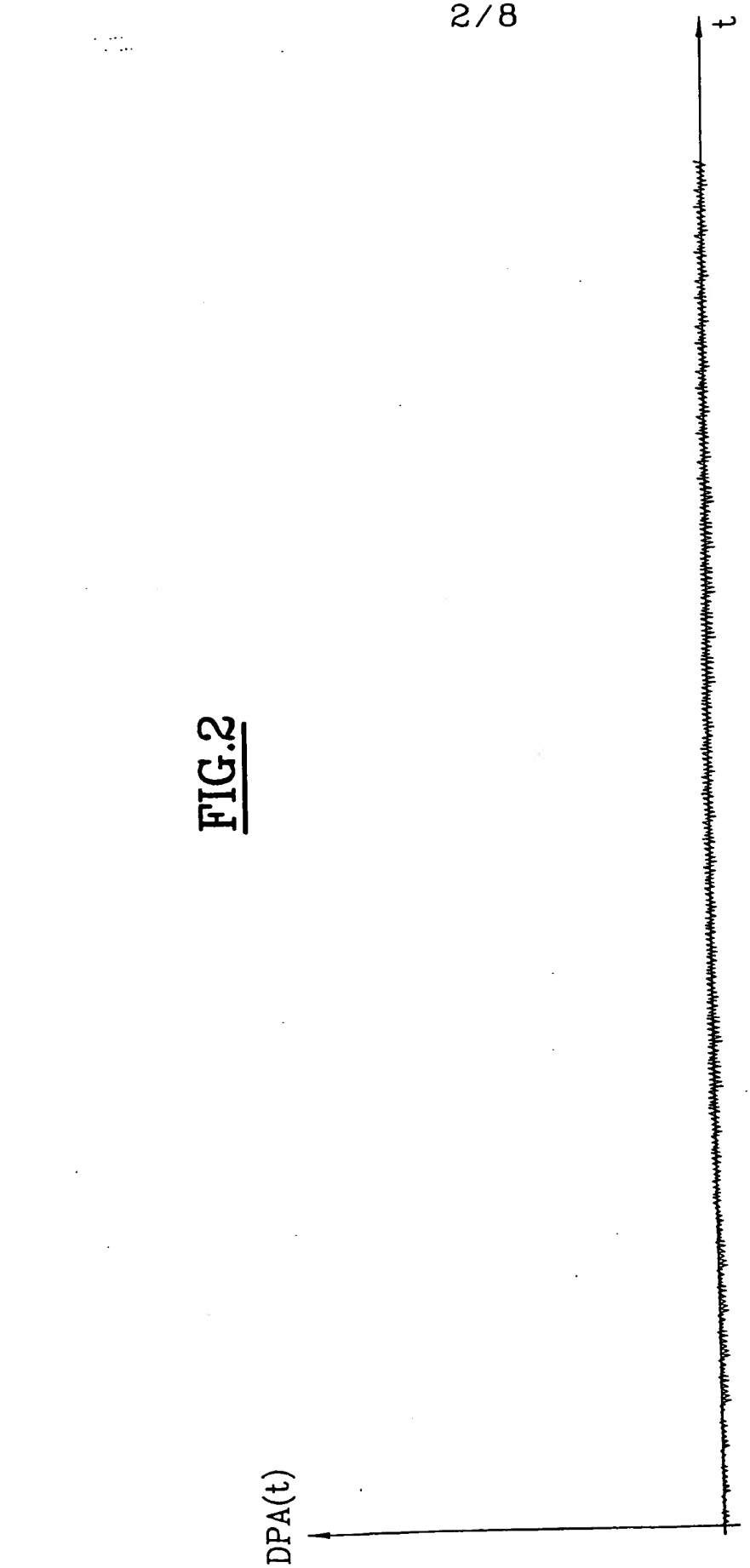


FIG.2



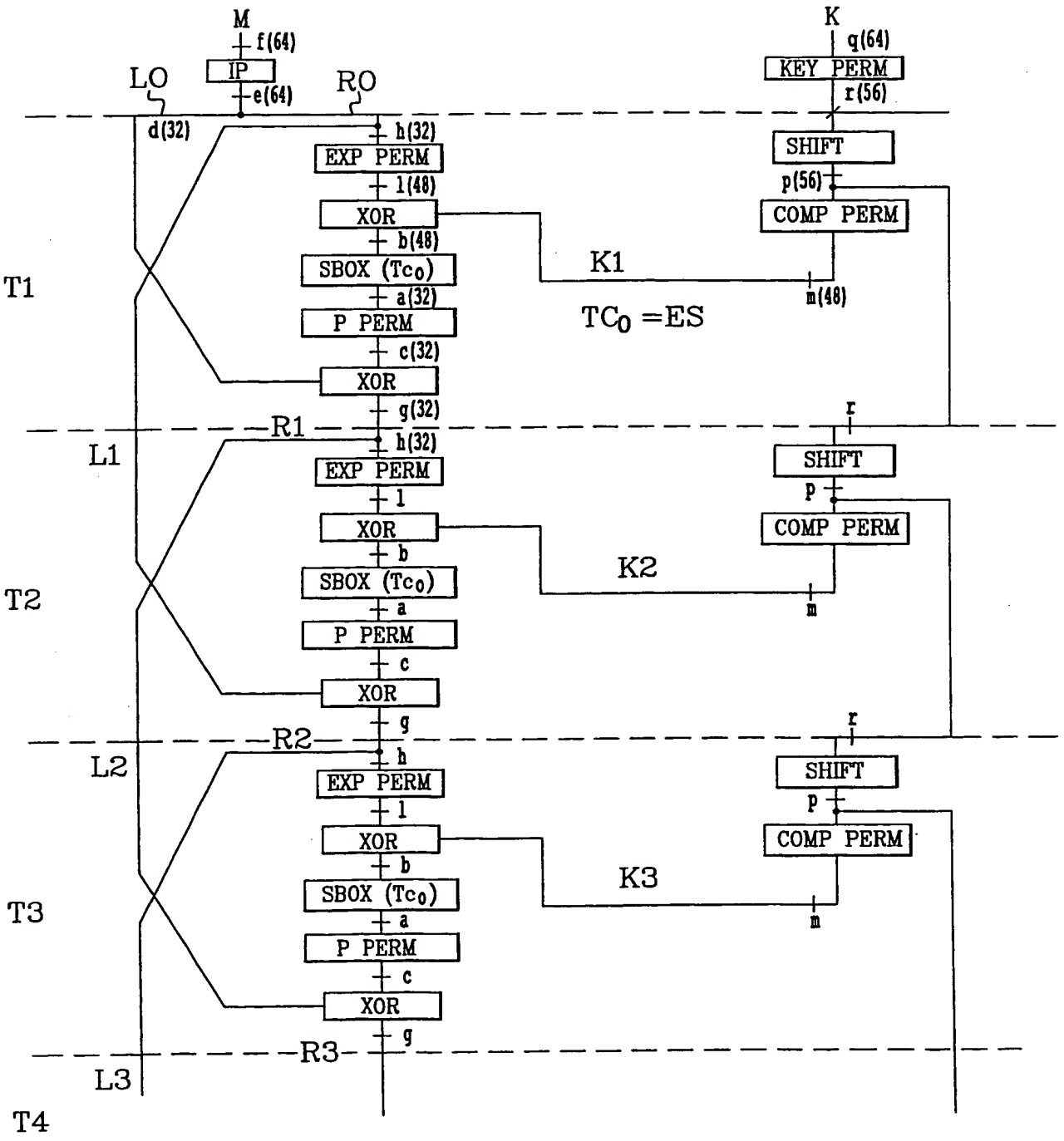

M
f(64)
IP
e(64)
LO
RO
d(32)
h(32)
EXP PERM
l(48)
XOR
b(48)
SBOX (Tc0)
a(32)
P PERM
c(32)
XOR
g(32)
K
q(64)
KEY PERM
r(56)
SHIFT
p(56)
COMP PERM
K1
m(48)
TC0 =ES
T1
R1
L1
T2
R2
L2
T3
R3
L3
T4
K2
K3

FIG.3

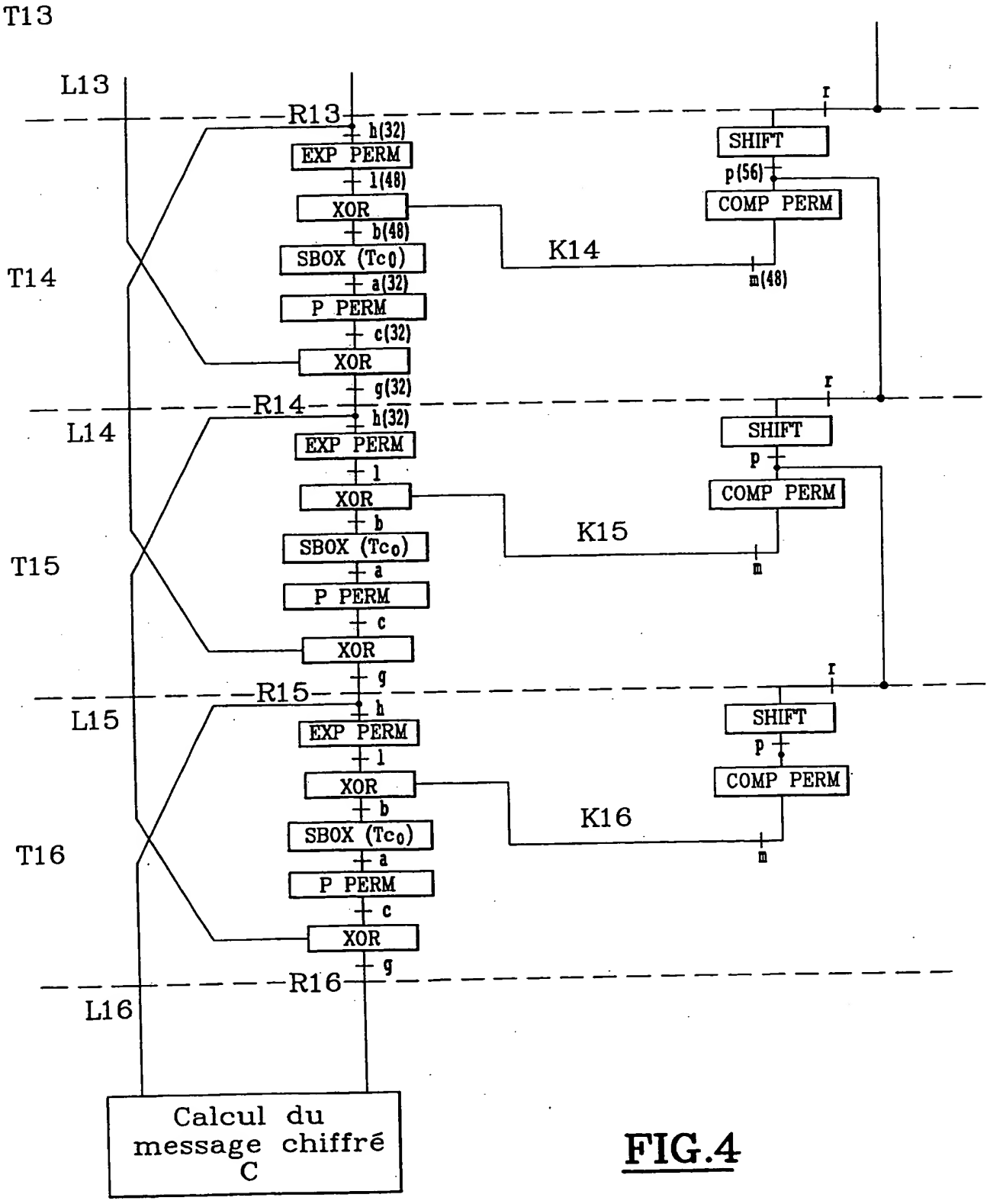

T13
L13
R13
h(32)
EXP PERM
l(48)
XOR
b(48)
SBOX (Tc0)
a(32)
P PERM
c(32)
XOR
g(32)
T14
R14
L14
h(32)
EXP PERM
l
XOR
b
SBOX (Tc0)
a
P PERM
c
XOR
g
T15
R15
L15
h
EXP PERM
l
XOR
b
SBOX (Tc0)
a
P PERM
c
XOR
g
T16
R16
L16
Calcul du message chiffré C
r
SHIFT
p(56)
COMP PERM
K14
m(48)
r
SHIFT
p
COMP PERM
K15
m
r
SHIFT
p
COMP PERM
K16
m

FIG.4

FIG.5

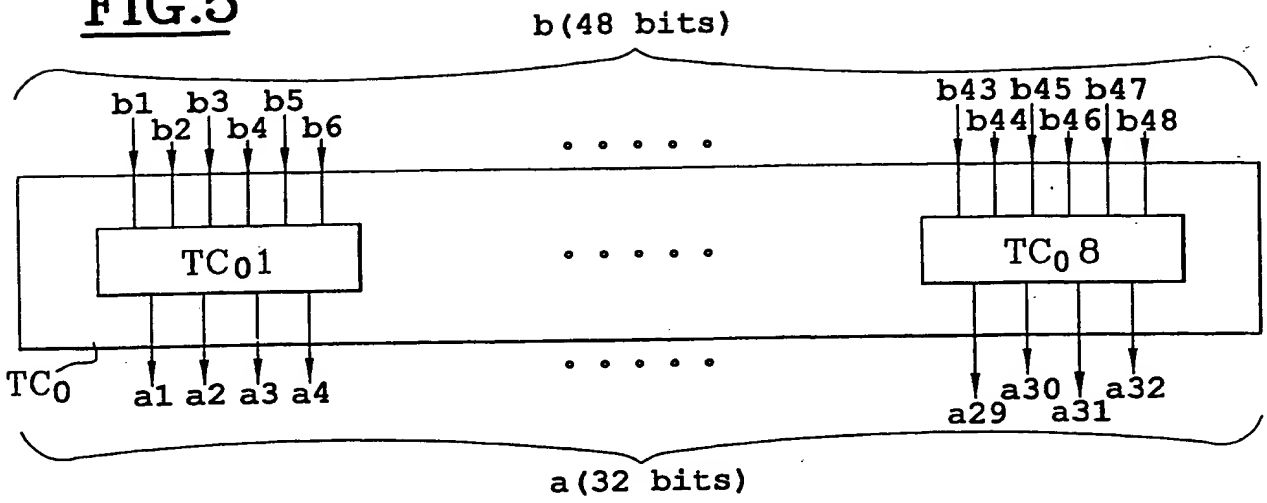


FIG.6

TC0 1

E1=b1b2b3b4b5b6	S1=a1a2a3a4
000000	1101
000001	0101
⋮	⋮
111111	1010

FIG.9

TC1 1

E1=b1b2b3b4b5b6	/S1=$\overline{a1a2a3a4}$
000000	0010
000001	1010
⋮	⋮
111111	0101

TC2 1

/E1=$\overline{b1b2b3b4b5b6}$	/S1=$\overline{a1a2a3a4}$
000000	0101
⋮	⋮
111110	1010
111111	0010

FIG.10

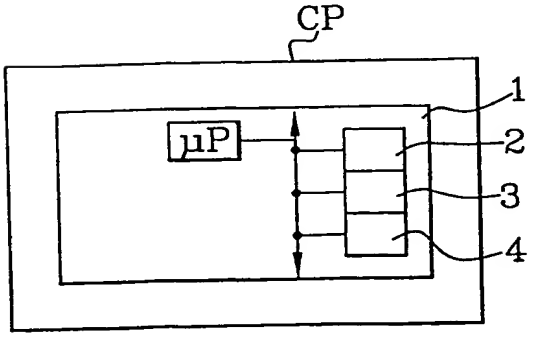


FIG.12

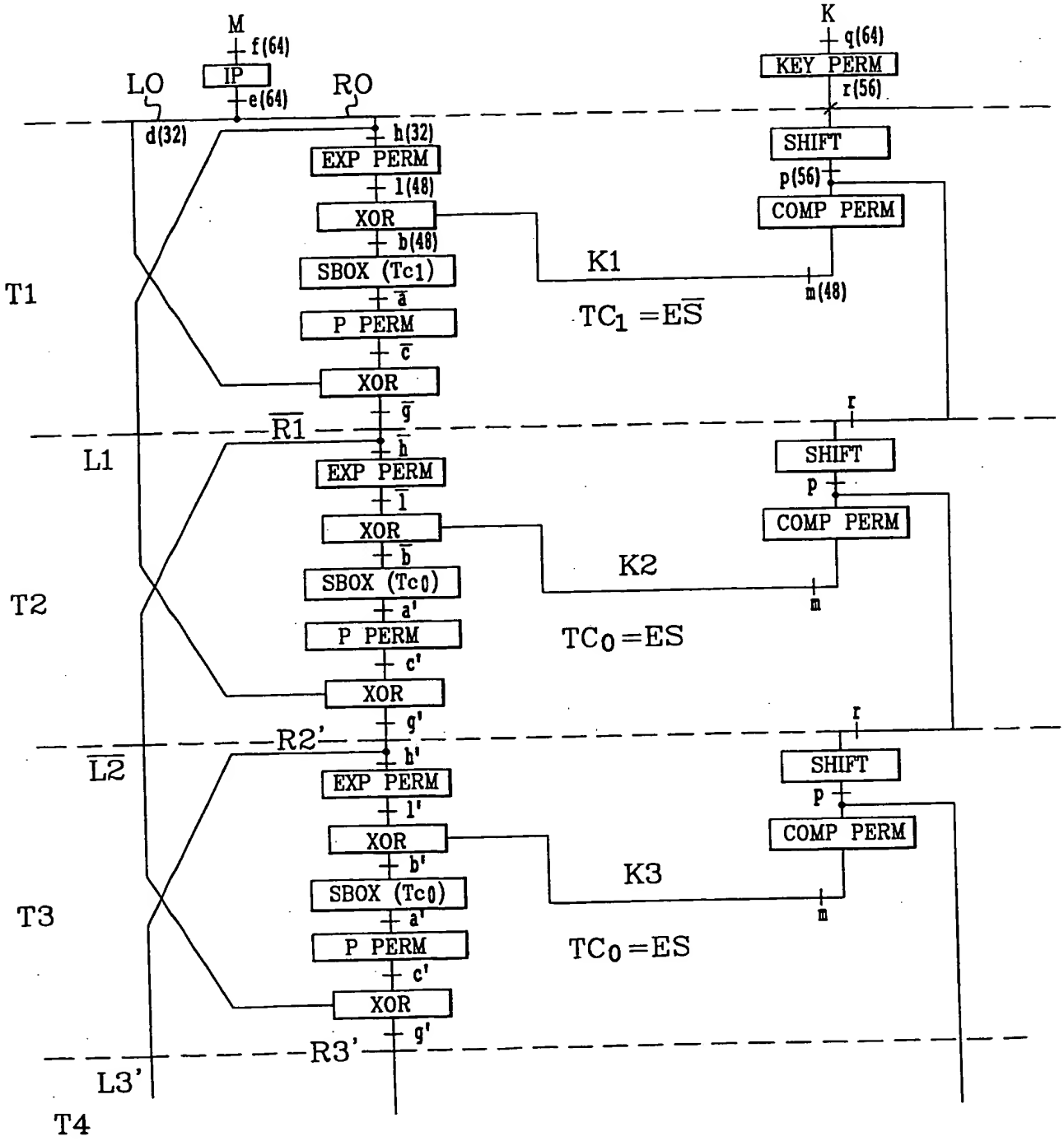

M
f(64)
IP
e(64)
LO
RO
d(32)
h(32)
EXP PERM
l(48)
XOR
b(48)
SBOX (Tc1)
a
P PERM
c
XOR
g
R1
L1
T1
K
q(64)
KEY PERM
r(56)
SHIFT
p(56)
COMP PERM
K1
m(48)
TC1 = ES
r
SHIFT
p
COMP PERM
K2
m
TC0 = ES
h
EXP PERM
l
XOR
b
SBOX (Tc0)
a'
P PERM
c'
XOR
g'
R2'
L2
T2
r
SHIFT
p
COMP PERM
K3
m
TC0 = ES
h'
EXP PERM
l'
XOR
b'
SBOX (Tc0)
a'
P PERM
c'
XOR
g'
R3'
L3'
T3
T4

FIG.7

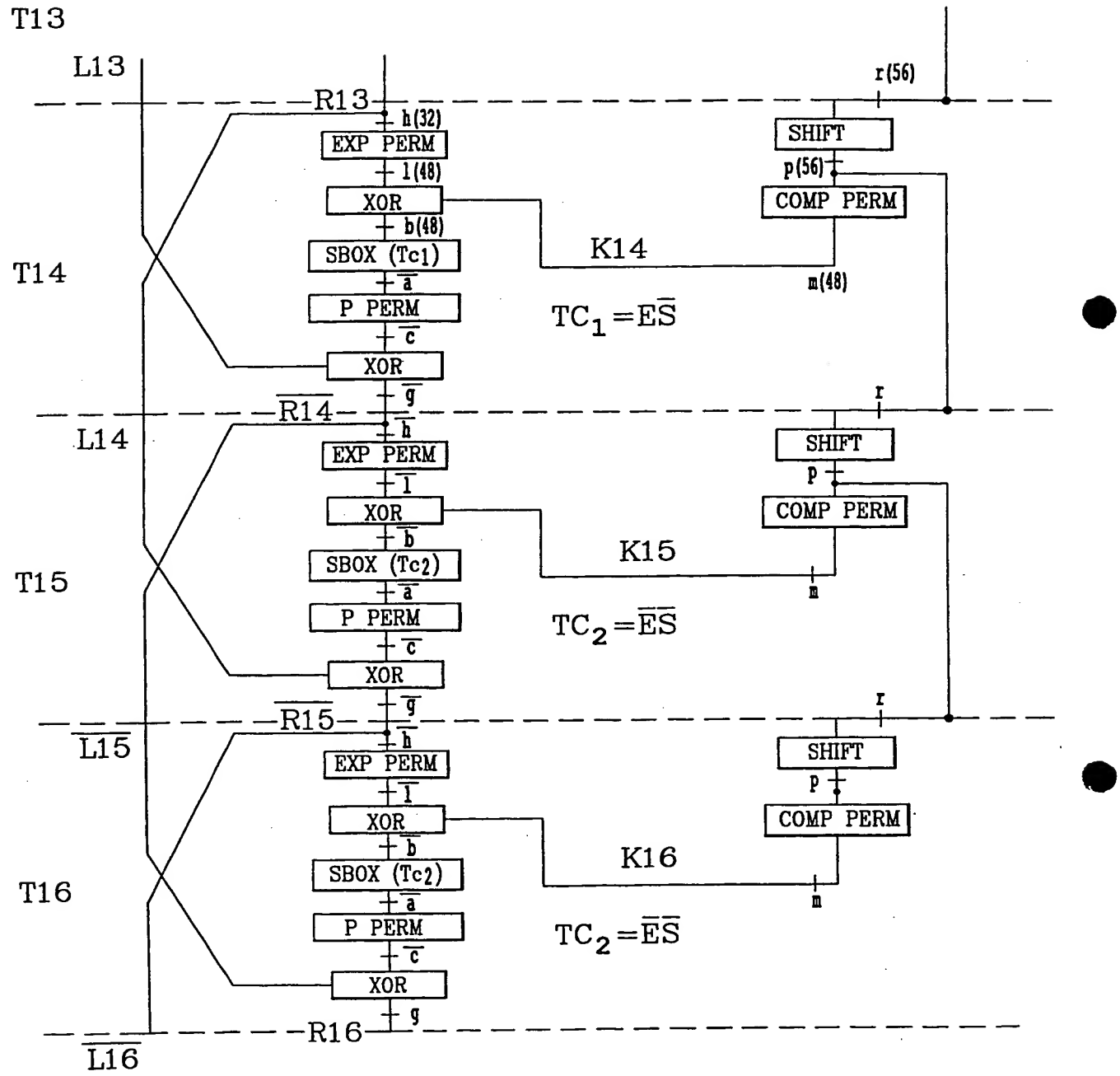

T13
L13
R13
h(32)
EXP PERM
l(48)
XOR
b(48)
SBOX (Tc1)
a
P PERM
c
XOR
g
r(56)
SHIFT
p(56)
COMP PERM
K14
m(48)
TC1 = ES
T14
R14
L14
h
EXP PERM
l
XOR
b
SBOX (Tc2)
a
P PERM
c
XOR
g
r
SHIFT
p
COMP PERM
K15
m
TC2 = ES
T15
R15
L15
h
EXP PERM
l
XOR
b
SBOX (Tc2)
a
P PERM
c
XOR
g
r
SHIFT
p
COMP PERM
K16
m
TC2 = ES
T16
R16
L16

FIG.8

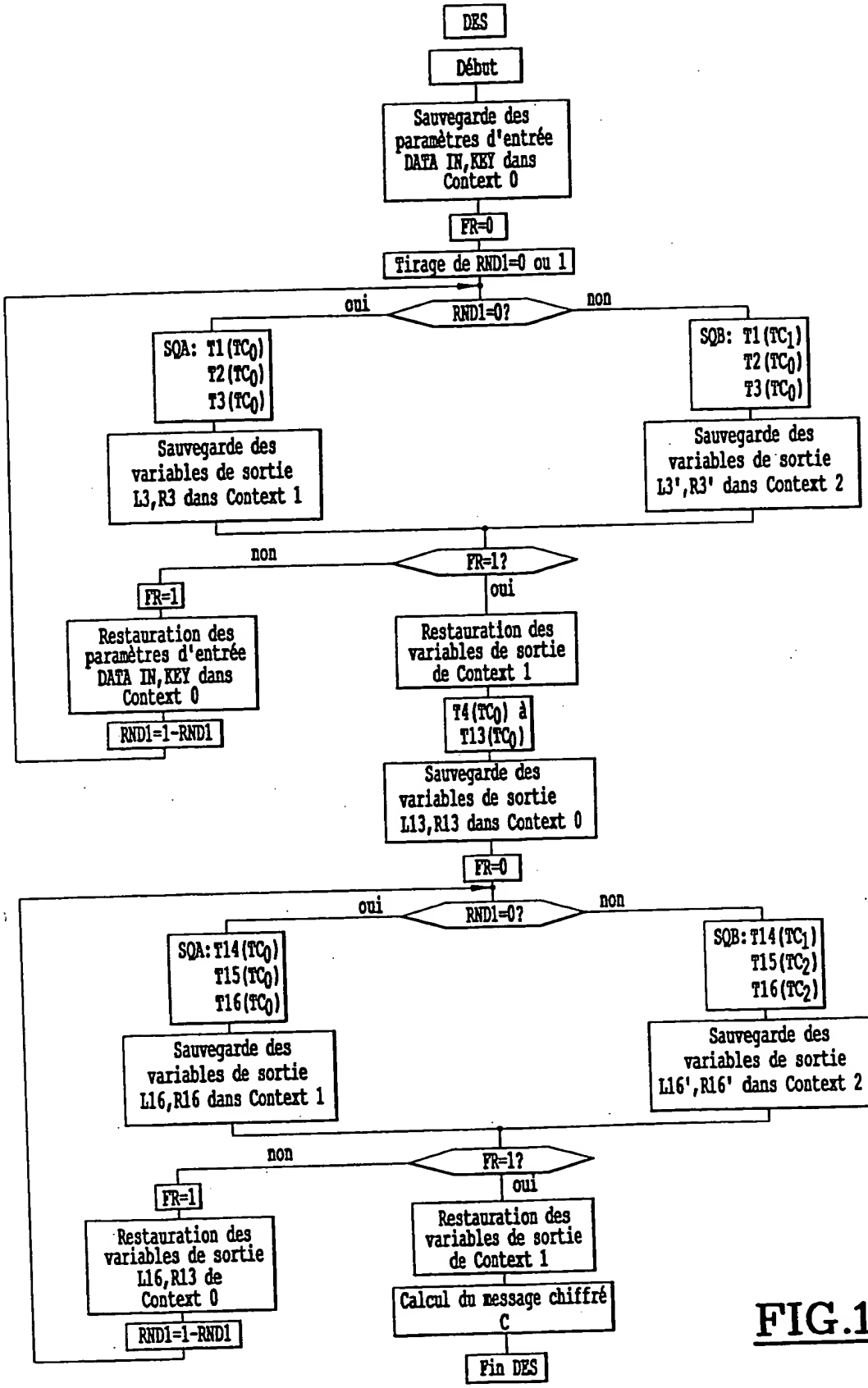

DES
Début
Sauvegarde des paramètres d'entrée DATA IN,KEY dans Context 0
FR=0
Tirage de RND1=0 ou 1
RND1=0?
oui
non
SQA: T1(TC0) T2(TC0) T3(TC0)
SQB: T1(TC1) T2(TC0) T3(TC0)
Sauvegarde des variables de sortie L3,R3 dans Context 1
Sauvegarde des variables de sortie L3',R3' dans Context 2
FR=1?
non
oui
FR=1
Restauration des paramètres d'entrée DATA IN,KEY dans Context 0
RND1=1-RND1
Restauration des variables de sortie de Context 1
T4(TC0) à T13(TC0)
Sauvegarde des variables de sortie L13,R13 dans Context 0
FR=0
RND1=0?
oui
non
SQA: T14(TC0) T15(TC0) T16(TC0)
SQB: T14(TC1) T15(TC2) T16(TC2)
Sauvegarde des variables de sortie L16,R16 dans Context 1
Sauvegarde des variables de sortie L16',R16' dans Context 2
FR=1?
non
oui
FR=1
Restauration des variables de sortie L16,R13 de Context 0
RND1=1-RND1
Restauration des variables de sortie de Context 1
Calcul du message chiffré C
Fin DES

FIG.11